

AMR Annotation: Special Topics

<http://tiny.cc/amrtutorial>

Congratulations!

You now know how to AMR **simple sentences**.

BUT: English is a wily opponent.

- Copulas, light verbs
- Derivational morphology
- Relational nouns
- Coordination & clausal connectives
- Modality
- Non-declarative sentences
- Questions
- Comparisons
- Quantification
- Subsets

Light semantics

- We try to eliminate purely grammatical words. E.g.:

- ▶ **copulas:** *I **am** happy*

(h / happy :domain (i / i))

- ▶ **light verbs:** *I'm **taking** a bath*

(b / bathe-01 :ARG0 (i / i))

Typical uses of inverse roles

- **Relative clauses:** *someone **who** sifts thistles*
- **Participles:** *thistle-sift**ing** person*
- **Derivational morphology:** *thistle sift**er***
- (p / person  :ARG0-**of** (s / sift-01
:ARG1 (t / thistle)))

Compositionality criterion

- We only “decompose” **derivational morphology** if a relative clause paraphrase is possible:
 - ▶ teach^{er} = person who teaches
(p / person :ARG0-of (t / teach-01))
 - ▶ profess^{or} ≠ person who professes
(p / professor)

Compositionality criterion

- Often core roles are available for modifiers:

- ▶ **math** teach**er** / teach**er** of **math**
= person who teaches math

(p / person :ARG0-**of** (t / teach-01
:ARG1 (m / **math**)))

- ▶ **math** profess**or** ≠ person who professes math

(p / professor :**mod** (m / **math**))

Compositionality criterion

- Sometimes it is difficult to draw a line, but we do our best:

- ▶ opin**ion** = thing that is opined

(t / thing :ARG1-of (o / opine-01))

- ▶ profess**ion** ≠ thing that is professed

(p / profession)

Hallucinating relations

- Sometimes we have to “hallucinate” a **relationship that the grammar underspecifies**.
 - ▶ e.g., **possessives** and **noun-noun compounds** can express many different kinds of relations

Relational nouns

- Special predicates for individual–group and individual–individual relations:

He is a pilot for TWA

I am your father

He is a TWA pilot

(h / **have-org-role-91**

:ARG0 (h2 / he)

:ARG1 (c / company

:name (n / name :op1 "TWA"))

:ARG2 (p / pilot))

(h / **have-rel-role-91**

:ARG0 (i / i)

:ARG1 (y / you)

:ARG2 (f / father))

Coordination & Clausal Connectives

| Example connectives | AMR treatment |
|--------------------------------|------------------------------|
| and | and |
| or | or |
| but | contrast-01 |
| because; due to; on account of | :cause |
| (in order) to; so (that) | :purpose |
| if | :condition |
| unless | :condition (... :polarity -) |
| although; despite | :concession |

Coordination

- The most common patterns:

X, Y, and Z

X, Y, or Z

X but Y

(a / and

:op1 X

:op2 Y

:op3 Z)

(o / or

:op1 X

:op2 Y

:op3 Z)

(c / contrast-01

:ARG1 X

:ARG2 Y)

- “and”, “or” take 2 or more conjuncts in sequence as :op#

Coordination



Coordination

- Rachael Ray finds inspiration in cooking her family and her dog.

```
(i / inspire-01  
  :ARG0 (c / cook-01
```

```
    )  
    :ARG1 (p / person :name (n / name :op1 "Rachael" :op2 "Ray"))))
```

Coordination

- Rachael Ray finds inspiration in cooking her family and her dog.

```
(i / inspire-01
  :ARG0 (c / cook-01
    :ARG0 p
    :ARG1 (a / and
      :op1 (f / family
        )
      :op2 (d / dog
        )))
  :ARG1 (p / person :name (n / name :op1 "Rachael" :op2 "Ray")))
```

Coordination

- Rachael Ray finds inspiration in cooking her family and her dog.

```
(i / inspire-01
  :ARG0 (c / cook-01
    :ARG0 p
    :ARG1 (a / and
      :op1 (f / family
        )
      :op2 (d / dog :poss p)))
  :ARG1 (p / person :name (n / name :op1 "Rachael" :op2 "Ray")))
```


Coordination

- Rachael Ray finds inspiration in cooking her family and her dog.

```
(i / inspire-01
  :ARG0 (c / cook-01
    :ARG0 p
    :ARG1 (a / and
      :op1 (f / family
        :ARG1-of (h / have-org-role-91
          :ARG0 p
          :ARG2 (m / member)))
      :op2 (d / dog :poss p)))
  :ARG1 (p / person :name (n / name :op1 "Rachael" :op2 "Ray")))
```

X's family = family of
which X is a member

Coordination

- Rachael Ray finds inspiration in cooking, her family, and her dog.

```
(i / inspire-01
  :ARG0 (a / and
    :op1 (c / cook-01
      :ARG0 p)
    :op2 (f / family
      :ARG1-of (h / have-org-role-91
        :ARG0 p
        :ARG2 (m / member)))
    :op3 (d / dog :poss p))
  :ARG1 (p / person :name (n / name :op1 "Rachael" :op2 "Ray")))
```

Coordination: shared core args

- We invited **and** then disinvited the students.

```
(a / and
  :op1 (i / invite-01
    :ARG0 (w / we)
    :ARG1 (s / student))
  :op2 (d / disinvite-01
    :ARG0 w
    :ARG1 s
    :time (t / then)))
```

Coordination: shared non-core args

- Yesterday we invited and then disinvited the students.

```
(a / and
  :op1 (i / invite-01
        :ARG0 (w / we)
        :ARG1 (s / student))
  :op2 (d / disinvite-01
        :ARG0 w
        :ARG1 s
        :time (t / then))
  :time (y / yesterday))
```

Coordination: copied predicates

- We invited the students **and** then the professors.

```
(a / and
  :op1 (i / invite-01
    :ARG0 (w / we)
    :ARG1 (s / student))
  :op2 (i2 / invite-01
    :ARG0 w
    :ARG1 (p / professor)
    :time (t / then)))
```

Modal Concepts

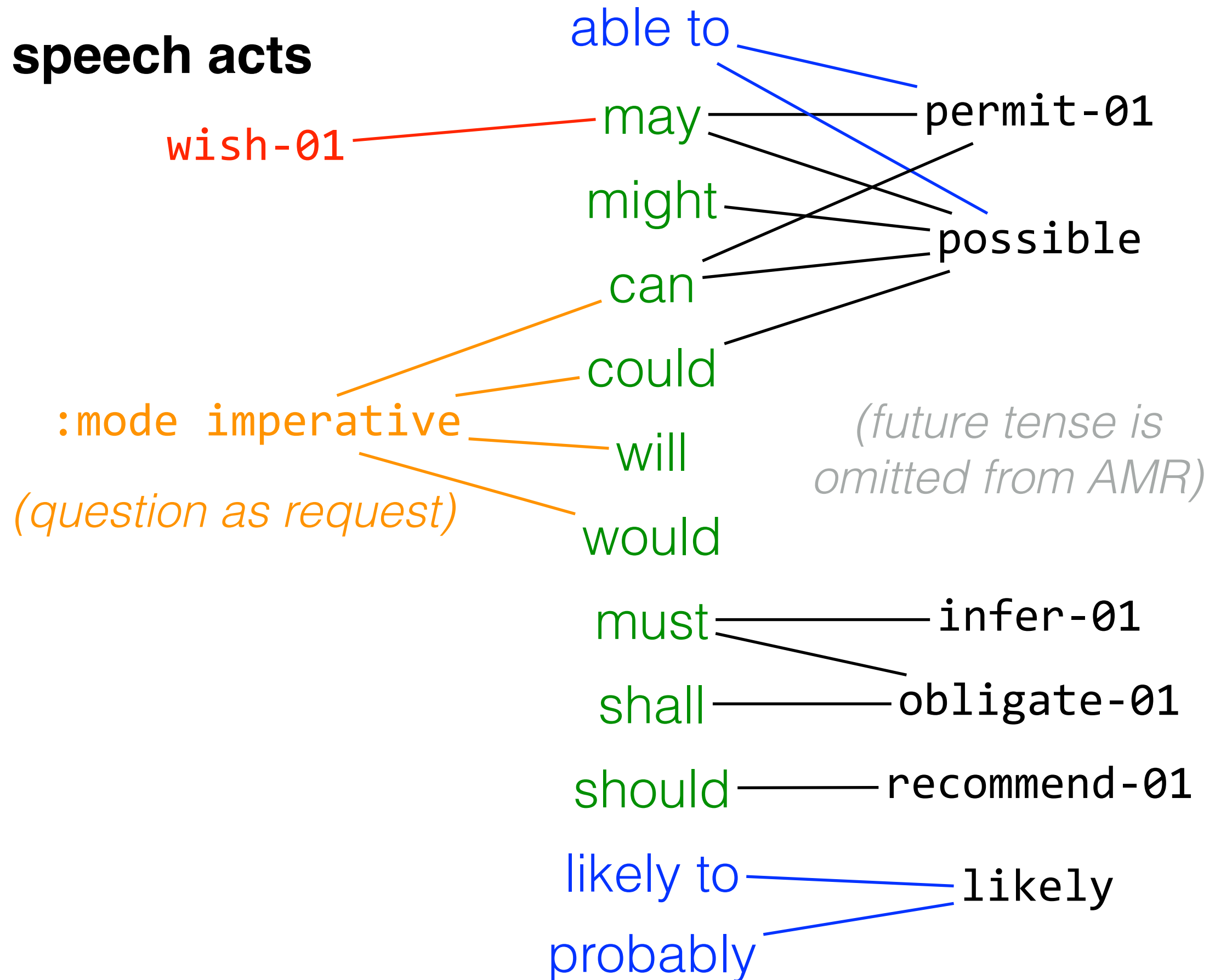
- You **can** leave.
You **may** leave.
It's all right for you to leave.

```
(p / permit-01
  :ARG1 (1 / leave-01
    :ARG0 (y / you))
```

- I **can** see Russia from my house!
I'm able to see Russia from my house!

```
(p / possible
  :domain (s / see-01
    :ARG0 (i / i)
    :ARG1 (c / country :name (n / name :op1 "Russia"))
    :location (h / house :poss i)))
```

speech acts



Sentence Types

| Type | AMR treatment |
|-------------------------------|---|
| indicative (declarative) | <i>(default)</i> |
| imperative (command) | <code>:mode imperative</code> (with you arg if implied subject) |
| interjection | <code>:mode expressive</code> |
| yes-no question | <code>:mode interrogative</code> |
| WH-question | <code>amr-unknown</code> |
| quotation without speech verb | <code>(s / say-01 :ARG0 <speaker> ...)</code> |
| vocative | <code>(s / say-01 :ARG2 <addressee> ...)</code> |
| polite (“please”, etc.) | <code>:polite +</code> |

Questions: yes-no

- Are you worried?

```
(w / worry-01  
  :ARG0 (y / you)  
  :mode interrogative)
```

Questions: wh

- Why worry? (What is the point of worrying?)

```
(w / worry-01  
  :ARG0 (y / you)  
  :purpose (a / amr-unknown))
```



Think of amr-unknown as an *in situ* question pronoun. Structurally, the AMR is the same as a declarative sentence.

- What's the problem?

```
(p / problem  
  :domain (a / amr-unknown))
```

- How many peppers did Peter Piper pick?

```
(p / pick-10  
  :ARG0 (p2 / person :name (n / name :op1 "Peter" :op2 "Piper"))  
  :ARG1 (p3 / pepper  
    :quant (a / amr-unknown)))
```

Comparison

have-degree-91

ARG1: attribute

ARG2: domain, entity
characterized by attribute

ARG3: degree itself

ARG4: compared-to

ARG5: consequence,
result of degree

- The treatment of comparative constructions is changing.
- Apples are redder than bananas.

old way:

```
(r / red
  :domain (a / apple)
  :degree (m / more)
  :compared-to (b / banana))
```

new way:

```
(r / red
  :ARG1-of (h / have-degree-91
    :ARG2 (a / apple)
    :ARG3 (m / more)
    :ARG4 (b / banana)))
```

Quantification

- two apples

```
(a / apple  
  :quant 2)
```

- a lot of apples

```
(a / apple  
  :quant (1 / lot))
```

- All apples are fruit.

```
(f / fruit  
  :domain (a / apple  
            :quant (a / all)))
```

Only explicit quantifiers
are included in the AMR.

- Apples are fruit.

```
(f / fruit  
  :domain (a / apple))
```

Sets

include-91 - “subset”

ARG1: subset (or member)

ARG2: superset

ARG3: relative size of subset compared to superset

- Special predicate **include-91** for explicitly mentioned sets
- (I ate) 5 of the 12 donuts

```
(d / donut :quant 5  
  :ARG1-of (i / include-91  
    :ARG2 (d2 / donut :quant 12)))
```

- 42% of the donuts

```
(d / donut  
  :ARG1-of (i / include-91  
    :ARG2 (d2 / donut)  
    :ARG3 (p / percentage-entity :value 42)))
```

Reification

be-located-at-91 -
“reification of :location”
ARG1: entity
ARG2: location

- *the man **at** the store*

(m / man :location (s / store))

- What about: *the man **always** **at** the store?*

- ▶ Need to “modify” the relation!

- ▶ Solution: Convert (“**reify**”) the relation w/ a special frame

(m / man

:ARG1-of (b / be-located-at-91

:ARG2 (s / store)

:time (a / **always**)))

Reification

be-located-at-91 -
“reification of :location”
ARG1: entity
ARG2: location

- Reification also allows a relational predicate to be focused.
- The man **is** **at** the store.

(b / **be-located-at-91**
:ARG1 (m / man)
:ARG2 (s / store))

- I think the man **is** **at** the store.

(t / think-01
:ARG0 (i / i)
:ARG1 (b / **be-located-at-91**
:ARG1 (m / man)
:ARG2 (s / store)))

Reification

be-located-at-91 -
“reification of :location”
ARG1: entity
ARG2: location

- Every role has a designated reification—either a verb frame or a special -91 frame.
 - ▶ have-purpose-91, have-polarity-91, have-part-91, ...
 - ▶ ~~have topic 91~~ concern-02
- These slides **are** about semantics.

(c / concern-02

:ARG0 (s / slide :mod (t / this))

:ARG1 (s2 / semantics))

Other Phenomena

- Many other patterns for specific phenomena are documented in the **AMR Dictionary**. E.g.:
 - ▶ *We'll eat **like** kings* → **resemble-01**
 - ▶ *(Banarescu et al., 2013)* → **publication-91**

AMR Data

Real Data

- Thus far: mostly made-up examples
- Real sentences tend to be longer, but AMRed using the same principles

Real Data

- We need to borrow 55% of the hammer price until we can get planning permission for restoration which will allow us to get a mortgage.

Real Data

control structure

- We need to borrow 55% of the hammer price until we can get planning permission for restoration which will allow us to get a mortgage.

Real Data

subset → include-91

- We need to borrow 55% of the hammer price until we can get planning permission for restoration which will allow us to get a mortgage.

Real Data

temporal connective

- We need to borrow 55% of the hammer price until we can get planning permission for restoration which will allow us to get a mortgage.

Real Data

- We need to borrow 55% of the hammer price until we can get planning permission for restoration which will allow us to get a mortgage.

modal

Real Data

- We need to borrow 55% of the hammer price until we can get planning permission for restoration which will allow us to get a mortgage.

deverbals → event predicates

Real Data

We need to borrow 55% of the hammer price until we can get planning permission for restoration which will allow us to get a mortgage.

(n / need-01
:ARG0 (w / we)
:ARG1 (b / borrow-01
:ARG0 w

)

)

Real Data

We need to borrow 55% of the hammer price until we can get planning permission for restoration which will allow us to get a mortgage.

```
(n / need-01
  :ARG0 (w / we)
  :ARG1 (b / borrow-01
    :ARG0 w
    :ARG1 (p / percentage-entity :value 55
      :ARG1-of (i / include-91
        :ARG2 (p2 / price
          :mod (h / hammer))))))
```

)

Real Data

We need to borrow 55% of the hammer price until we can get planning permission for restoration which will allow us to get a mortgage.

```
(n / need-01
  :ARG0 (w / we)
  :ARG1 (b / borrow-01
    :ARG0 w
    :ARG1 (p / percentage-entity :value 55
      :ARG1-of (i / include-91
        :ARG2 (p2 / price
          :mod (h / hammer))))))
  :time (u / until
    :op1 (p3 / possible
      :domain (g / get-01
        :ARG0 w
        :ARG1 (p4 / permit-01
          :ARG1 (p5 / plan-01)
          :purpose-of (r / restore-01)
          :ARG0-of (a / allow-01
            :ARG1 (m / mortgage-01
              :ARG0 w)))))))))
```

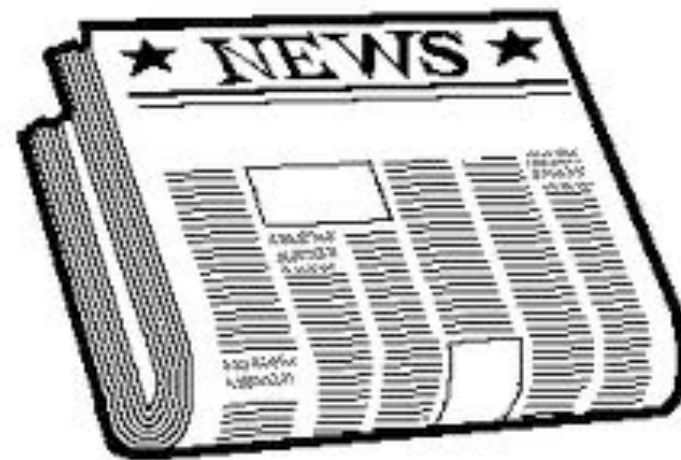
Datasets

AMR Bank: *The Little Prince*

(novel—English translation)

LDC Releases

(news, discussion forums, ...)



+



= 15k AMRs

(with more to come!)

Datasets: Details

- AMR Bank (Release 1.4; <http://amr.isi.edu/download/amr-bank-v1.4.txt>)
- English translation of The Little Prince, freely downloadable. 1,500 AMRs
- AMR Public Release 1.0 (LDC2014T12): largest public release w/ 13,051 AMRs
- DEFT Release 3 (LDC2013E117): evaluation data in Flanigan et al. 2014, Wang et al. 2015.
- DEFT Release 4 (LDC2014E41): largest release w/ 18,779 AMRs total
- DEFT Release 5 (Sep. 2015) will include wikification, (pretty much) no directed cycles
- Small (100-AMR) sets of Czech and Chinese AMRs have been annotated.
- Vanderwende et al. (2015) data to appear: several languages, automatically converted from logical forms
- PropBank will soon all be converted to AMR style (mapping nominalizations to verbs, etc) and re-released.

AMR
vs.
Other Formalisms

AMR Strengths

- **abstracting away from morphological & syntactic variability**
- predicate-argument structures
 - ▶ core + non-core roles
- named entities & values
- coreference (w/in sentence)
- modality

AMR Limitations

- no “deep” **lexical semantics**
 - *fruit/berry, buy/sell, kill/die* are formally unrelated
- no deep treatment of **quantification & scope**
- (almost) no **information structure**
- nothing across sentences in a **discourse**...yet

Design Decisions

- AMR annotations are **not tied to individual words or any syntactic derivation**
- **Practicality for human annotators** is primary
 - ▶ AMR makes no compromises for (current) algorithms
- **Single structure** rather than many layers
- Extensive **documentation and tool support**

Comparison - Semantic Roles

AMR: 70+ non-core roles, many verb-sense specific roles
(up to 5 args/roleset, more than 10,000 rolesets)

FrameNet: large inventory of frame-specific roles

VerbNet: inventory of thematic roles

Groningen Meaning Bank: VerbNet inventory

Most others: small inventory of roles (agent, theme, etc.)

Comparison - Sense Lexicon

Groningen Meaning Bank: (automatic) WordNet synsets

FrameNet/UCCA: Mark senses by frame/script, not lemma

AMR /PropBank: coarse-grained senses (get high ITA)

Prague Dependency TB: valency lexicon rolesets

Most others: undisambiguated concepts as predicates

Comparison - Entities

AMR: Rich named entity ontology (100+ types), wikification

GALE/Ontonotes Annotations: 29 types, 64 subtypes

Groningen Meaning Bank: 7 NE types

Domain-specific (ACE/UMLS/etc.): rich; not all entities

Others: no entity typing

Comparison - Alignment with text

Deepbank; Groningen Meaning Bank: Semantics linked up to a theory of its derivation from syntax (HPSG; CCG)

PropBank, Semantic Treebank: grounded in PTB

Most others: Some link to words in sentence

AMR: No alignment to text (plan to release a few alignments)

Comparison - Logic/Scope/Entailments

Deepbank; Groningen Meaning Bank: Semantics grounds out in logical formalisms (DRT and MRS, respectively)

AMR entailment: linkage between its lexicon and VerbNet may allow rich decomposition

AMR scope: No scope of quantification

Comparison - Size and Quality

AMR: 18,779 sentences, goes beyond newswire, fully manual

Prague Dependency TB: WSJ in Czech and English, manual

Deepbank; Groningen Meaning Bank: Large; automatic parses with human correction/feedback.

UCCA: fully manual, 160k tokens

Rich semantic systems with little affiliated data: TMR, LCS,

...